Available online at www.sciencedirect.com

ScienceDirect

Computer Communications 31 (2008) 1301–1317

computer communications

www.elsevier.com/locate/comcom

Available bandwidth estimation and admission control for QoS routing in wireless mesh networks

Mesut Ali Ergin [a,*], Marco Gruteser [a], Lin Luo [a], Dipankar Raychaudhuri [a], Hang Liu [b]

[a] *WINLAB, Electrical and Computer Engineering Department Rutgers, The State University of New Jersey,*
671 Route 1 South, North Brunswick, NJ 08902-3390, USA
[b] *Corporate Research Lab, Thomson Inc., 2 Independence Way, Princeton, NJ 08540-6620, USA*

Available online 26 January 2008

Abstract

This article presents an integrated admission control and routing mechanism for multi-rate wireless mesh networks. Admission control depends on precise estimates of available bandwidth at involved nodes and the bandwidth consumption required by a new flow. Estimating these parameters in wireless networks is challenging due to the shared and open nature of the wireless channel. Existing available bandwidth estimation techniques do not accurately consider interference from neighboring nodes and flow bandwidth requirement estimates or act overly conservative, restricting opportunities for parallel transmission due to spatial reuse. We propose the DCSPT method for available bandwidth estimation, based on dual carrier sensing with parallel transmission awareness. We also introduce a packet probing-based available bandwidth estimation method, suitable for legacy device implementations, and verify it experimentally. These techniques are integrated in an admission control mechanism designed for a hop-by-hop routing protocol (LUNAR), enabling alternate route identification when shortest paths are congested. Our protocol uses temporal accounting to enable bandwidth estimation across links using different bit-rates. Simulation results demonstrate that our admission control mechanism can effectively control the traffic load while considering parallel transmission opportunities, leading to cumulative system throughput improvements up to 80% compared to more conservative approaches. We further show that additional gains in system throughput come without significant cost in terms of packet delivery ratio or end-to-end delay and discuss our implementation experience on the ORBIT wireless research testbed.

Keywords: Wireless mesh networks; Available bandwidth estimation; Admission control; QoS routing

1. Introduction

Today, wireless LANs have been very successful in extending wired LANs with a last hop solution. However, untethered communication's full blessings are not limited to single-hop reach of wireless access points. Autonomous nature of wireless mesh networks (WMNs) makes them easy-to-maintain while providing robust and reliable extension to wired infrastructure [1]. Distinctive applications of IEEE 802.11-based WMNs comprise streaming of audio and video content between entertainment and surveillance devices. While we expect range and reliability improvements from multi-hop communication for these applications, WMNs still struggle to find effective ways of providing low delays and packet loss under impairments due to serious interference, uncoordinated medium access, and sub-optimal route selection. Effects of these impairments are disastrous when WMN operates close to congested-state. Fortunately, admission control [2] can prevent the network from reaching saturation by rejecting new data flows, if sufficient bandwidth is not available.

Providing admission control in wireless networks is particularly challenging because of the shared and open nature of wireless channel. Unlike wired networks, each node in the WMN might have a different view of the current channel usage, based on its unique position in space. In addi-

[*] Corresponding author. Tel.: +1 862 368 6620; fax: +1 512 628 3998.
E-mail addresses: ergin@winlab.rutgers.edu (M.A. Ergin), gruteser@winlab.rutgers.edu (M. Gruteser), clarylin@winlab.rutgers.edu (L. Luo), ray@winlab.rutgers.edu (D. Raychaudhuri), hang.liu@thomson.net (H. Liu).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.01.027

tion, the contention range of a wireless transceiver reaches beyond its direct communication range, making it difficult to determine whether a newly admitted flow affects existing flows at neighboring nodes.

This paper presents an admission control mechanism for WMNs, integrated with routing. Because mesh networks often provide multiple possible paths between a source-destination pair, integrating admission control into the routing protocol enables the identification of alternate routes if the shortest path is congested. Our key contributions with this paper include:

- An accurate available bandwidth estimation method (DCSPT) that can account for interference from carrier sensing neighbors without being overly conservative (i.e. parallel-transmission aware).
- A practical available bandwidth estimation method based on packet probing (PPRCH) that only requires software changes from commodity wireless mesh networking equipment.
- An admission control mechanism that can be used in multi-rate WMN environments, that is, channel utilization and sustainable bit-rate is considered together when calculating available bandwidth.
- Protocol extensions to exchange topology information needed for proposed admission control mechanism in hop-by-hop routing protocols such as AODV or LUNAR. Support for detecting and recovering from QoS violations.

The rest of the paper is organized as follows: Section 2 visits the literature related to our work and Section 3 covers the necessary background on link self-interference and available bandwidth estimation. Our available bandwidth methods are outlined in Sections 4 and 5 describes how admission control decision is made and integrated into the routing protocol. Performance evaluations are provided in Section 6 and some of our experiences from prototype implementations are given in Section 7. Finally, concluding remarks are provided in Section 8.

2. Related work

Quality-of-service (QoS) support for wireless mesh networks can be achieved though flow prioritization techniques or traffic shaping, but once a network approaches saturation, QoS for the existing flows can only be maintained through admission control (AC) mechanisms. These mechanisms, such as [3–6], allow a new flow onto the network only if sufficient bandwidth resources are available. Thus, available bandwidth (AB) estimation constitutes a critical component of all AC mechanisms. Many active (i.e. intrusive) and passive (i.e. non-intrusive) *wired-network techniques* [7–9] have been proposed for estimating end-to-end available (or achievable) bandwidth for the Internet. Unfortunately, many of the wired network bandwidth estimation methods are not directly applicable to wireless mesh networks, because, consumption of a certain amount of resource from the shared wireless medium is not easily detectable at all other affected mesh nodes. For wireless networks, a group of researchers [5,10,11] suggest using locally decodable packets for AB estimation. This approach, however, fails to account for resource consumption by interfering signals that do not result in a successfully received packet. Other AB estimation techniques using delay measurements [12,13] leave out practical performance evaluations under carrier-sensing neighbor interference. The most relevant to our approach with packet probes, *AdHoc Probe* [14], exclusively aims to measure link capacity under no background traffic, hence, it has limited value to AC in wireless mesh networks.

Similarly, solutions exist for AC in the wired domain [15] (such as IntServ/RSVP and DiffServ [16]), unfortunately they are not designed to address the issues of wireless ad hoc and mesh networks [17]. In some wireless-specific proposals, AB under-estimation due to the decodable packets approach and overlooking interference from other nodes remain a major drawback [5,18,19]. Another resource reservation algorithm, AQOR [6], fails to accurately estimate resource consumption at all contending nodes within carrier sensing range, when making AB calculations. A recent analytical model study on AC targets specifically mesh networks, however, specifics of its AB estimation method, interference-awareness, and protocol implementation are unknown [20]. The authors of [21] evaluate effects of two different interference approximation methods on a wireless mesh network VoIP *call admission control system* performance. The most related study to our work, CACP, is presented in [3]. The authors introduce concept of *c-neighborhood* AB, and their AC scheme considers interference from carrier sensing neighbors. We build on CACP and develop an AC mechanism that can operate in a multi-rate environment. Our scheme provides more accurate AB estimation by considering parallel transmissions opportunities. Also, our protocol design is based on a partial topology-sharing hop-by-hop routing protocol, instead of source routing.

3. Background

In this section, we discuss the necessary background information on how we estimate link utilization of a single-hop flow. Then, we leverage this information to estimate local utilization due to a multi-hop flow, which changes at each hop due to the effect of self-interference [3] in the mesh network.

3.1. Self-interfering flows

Given the bandwidth requirement of a flow, we calculate link utilization requirement of a mesh node by considering basic contention-based access mechanism of IEEE 802.11 DCF [22]. Readers are assumed to be familiar with the basics of IEEE 802.11 DCF, and referred to [23] for more information. According to the DCF channel access mechanism, single-hop channel occupation duration of a data packet, T_{occup}, can be expressed as

$$T_{\text{occup}} = 4T_{\text{PLCP}} + T_{\text{difs}} + T_{\text{backoff}} + T_{\text{rts}} + T_{\text{cts}} + \frac{L}{B} + T_{\text{ack}} + 3T_{\text{sifs}}, \quad (1)$$

where L is the length of the data packet including MAC header, T_{PLCP}, T_{rts}, T_{cts}, and T_{ack} are time taken by PHY layer *PLCP* header, *RTS*, *CTS*, and *ACK* packet transmissions, respectively. T_{sifs} and T_{difs} denote short and DCF inter-frame spacings of the IEEE 802.11 standard. B is the link rate used by the transmitting node. T_{backoff} denotes the random backoff duration prior to the transmission. Due to our focus on regulating the traffic load on the channel and providing QoS guarantees to admitted flows, we assume the transmission failures due to collisions (i.e. frequently seen when operating close to full utilization) are negligible. Consequently, backoff duration T_{backoff} can be approximated by its expectation $E[T_{\text{backoff}}] = \frac{CW_{\min}}{2} \times T_{\text{slot}}$. Expression (1) can then be re-written as the summation of a link-rate/frame-size dependent term, $T_{\text{data}} = \frac{L}{B}$, and a link-rate/frame-size independent term, T_{oh}, such that

$$T_{\text{occup}} = T_{\text{data}} + T_{\text{oh}}. \quad (2)$$

If the source of a flow generates a packet every $\frac{1}{R}$ s on average, link utilization requirement due to this flow on a single-hop, ρ_{req}, can be expressed as

$$\rho_{\text{req}} = \frac{T_{\text{occup}}}{\frac{1}{R}} = R \times (T_{\text{data}} + T_{\text{oh}}). \quad (3)$$

ρ_{req} values greater than 1 indicates an unattainable flow rate being offered to the link. Also, above equation for ρ_{req} is dependent on the link rate due to T_{data}, suggesting that a flow would require different utilization values from nodes using different rates for transmission. We will elaborate more on the details of muti-hop forwarding in the following section.

3.2. Link utilization requirement along forwarding path

Aggregate link utilization change along the forwarding path of an admitted flow depends on the spatial distribution of the nodes forwarding this flow and the link-rates used by them. Some of the nodes on the forwarding path compete for the channel while transmitting packets of the same flow. This has been referred to as *self-interference* or *intra-flow contention*. As an example, consider the residential wireless mesh network in Fig. 1. In this network, the flow originating from the node C1 is being forwarded to the node C4. The dashed circle in the center represents node A's carrier-sensing (CS) range (CSR_{A}). An ongoing transmission originating from anywhere in this range (i.e., from nodes N1, N2, N3, and N4) would defer transmissions from node A, reducing node A's chances of channel access. When calculating utilization of the link A → N3 due to the flow being forwarded across the network, utilization of links N1 → N2, N2 → A, N3 →N4, and N4 → C3 must also be considered.

Assume the link rates used on N1 → N2, N2 → A, A → N3, N3 → N4, and N4 → C3 are B_1, B_2, B_3, B_4, and B_5, respectively. Then, the aggregate link utilization change required by this flow, from the perspective of node A, can be calculated as $\sum_{i=1}^{5} R \cdot \left(\frac{L}{B_i} + T_{\text{oh}}\right)$. By generalizing utilization expression for an outbound link having N_{cont} contending forwarder nodes within the transmitters CS-range, we get

$$\rho_{\text{aggr}} = \sum_{i=1}^{N_{\text{cont}}+1} R \cdot (\frac{L}{B_i} + T_{\text{oh}}). \quad (4)$$

Note that N_{cont} does not include the flow's final destination, if that destination falls within transmitters CS-range. This calculation exploits the fact that when actively transmitting, any of the nodes N1, N2, N3, and N4 would interfere with node A. This approach in its current form, similar to the one in [3], might be overly conservative, neglecting likely opportunities of simultaneous transmissions (such as from N1 and N4) thus under-estimating available channel time. We will consider parallel transmission opportunities as we discuss available bandwidth estimation via DCSPT method in Section 4.1.

Determining and maintaining exact number of nodes in a given transmitters CS-range, N_{cont}, is difficult in wireless mesh networks for a number of reasons. Being within a node's CS-range does not imply direct communication abilities with the node. This is because for all practical purposes, CS-thresholds on radios are tuned to conservatively low levels in order to protect neighboring receivers from destructive interference. To capture this phenomenon, for example, the ns-2 simulator defines the CS-range in IEEE 802.11 as approximately twice the transmission range at 2 Mbits/s bit rate [24]. Lack of direct communication ability makes it impossible to rely on single-hop observation-based N_{cont} estimations. Signal strength variations due to fading conditions in the environment might also cause some nodes to go back and forth in terms of being a carrier sensing neighbor to others. Besides, even limited mobility in the network makes estimating N_{cont} more difficult, due to the resulting topological changes.

In this study, we approximate N_{cont} by counting neighbors that are reachable within *k-hops* that lie on the forwarding path of the flow. In a single-rate IEEE 802.11 network, we can easily use k as the number of non-destination nodes within a flow's $\left\lceil \frac{CS-range}{TX-range} \right\rceil$ hop reach. However, k in a multi-rate network has to be treated differently. We use of the number of hops to approximate a fixed CS-range, however, the number of hops within a fixed distance depends on the particular rate being used. This is due to varying communication ranges of different modulation schemes with different SINR budget requirements [25]. In the multi-rate case, an approach to exchange per-hop rate information between the nodes along the forwarding path

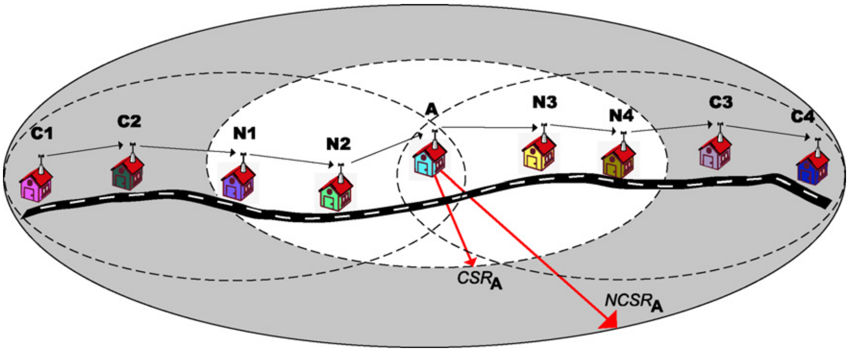


Fig. 1. An example self-interfering flow along its forwarding path in the mesh network. Transmissions from nodes N1, N2, N3, and N4 compete with the transmissions from node A.

is necessary and it can be easily adopted to LUNAR hop-by-hop route establishment mechanism which we will detail in Section 5. Once a node learns about the rates its neighbors used for a particular flow, a value for k can be estimated via a lookup table of distance-rate relations, assuming an appropriate propagation model. To avoid complexity in our evaluations, we have used a single-rate network of 2 Mbits/s, where k has been used as 2.

3.3. Available bandwidth estimation

Available bandwidth estimation is the most critical function in all admission control mechanisms that are applicable to wireless mesh networks. We define *available bandwidth* (AB) on a wireless mesh node as the rate of *additional layer-2 traffic* that can be transmitted (or relayed) from a node (towards a certain destination) without causing degradation of service to other ongoing flows in the network. *Maximum achievable bandwidth* (MAB), on the other hand, is defined as the maximum attainable rate of traffic that can be transmitted (or relayed) from a node (towards a certain destination), when *no other competing traffic exists* in the network. MAB is an indication of potential link capacity and in this study, we are interested in AB estimation, since it relates directly to admission control for supporting QoS on wireless mesh networks.

AB in a wireless mesh networking context, has to be treated *per destination* that is fundamentally different from the AB in a wired networking context. Objective of an AB estimation method is to accurately assess the percentage of the air-time that is available for additional traffic, namely complementary of link utilization $(1-\rho)$. This utilization estimate must be considered together with the sustainable link rate towards a certain destination while obtaining the AB towards that destination. For example, in an IEEE 802.11b network, $(1-\rho)$ value of 0.5 for a given link might allow for an additional 5.5 Mbits/s traffic (ideal PHY layer rate) towards a very high SNR receiver that can be communicated at 11 Mbits/s. The same utilization might only allow for 500 Kbits/s towards another receiver that sees a very low SNR, thus forcing the link to operate at 1 Mbits/s. Note that utilization here reflects channel occupancy from all interfering links in the region of interest.

4. Methods for AB estimation in wireless mesh networks

In the following subsections, we will discuss two AB estimation methods that we propose for wireless mesh networks. First, we explain *Dual-Carrier Sense with Parallel Transmission-awareness* (DCSPT) method [26]. This method requires explicit hardware support for carrier-sensing functionality. Then, an alternative that can be implemented in software, *Packet Probing with RTS/CTS Handshake* (PPRCH) method [27] follows.

4.1. Dual-carrier sense with parallel transmission-awareness (DCSPT)

DCSPT [26] builds on protocols that exploit adjustable carrier-sensing (CS) thresholds (a.k.a. CCA-threshold) of the wireless transceivers [4,3]. Their fundamental idea is that the extent to which a node is aware of its surrounding transmissions can be changed by adjusting its CS-threshold. With the proper admission control mechanism, this awareness would help preventing a node from accepting flows that might violate QoS constraints of some of its neighbors. The main contribution of DCSPT lies in showing that protocols that use adjustable CS-thresholds, such as [3], are overly conservative; they under-estimate AB because they do not account for several transmissions occurring in parallel due to spatial reuse. This reuse is typically observed between the transmissions of the same flow, as well as between the transmissions of different flows crossing the network. DCSPT method takes into account the opportunities of parallel transmissions and yields more accurate AB estimation.

To summarize the basic operation of DCSPT, let us consider the example mesh network given in Fig. 2. Dashed ellipse in the figure shows the CS-range of node C (CSR_C). When node C attempts to initiate a flow to node D, it can measure local channel utilization by channel-busy time according to its regular CS-threshold. This channel utilization, however, would ignore the air-time resource

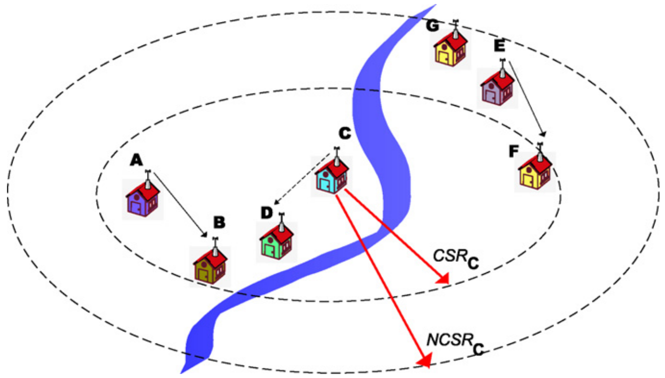


Fig. 2. Illustrating effects of CS-threshold on AB-estimation.

consumption due to the flow from node E to node F. In other words, by initiating a new flow, node C might affect the flow from node E to node F, unknowingly. Even using the optional virtual carrier sensing mechanism in the IEEE 802.11 (RTS/CTS and NAVs) would not help solving this issue since node F is close to the edge of CSR_{C}, making node F's CTS packets invisible at node C. On the other hand, by lowering its CS-threshold, node C could have accounted for the transmission from node E in the AB estimation process.

We adopt a two-level CS-threshold adjustment scheme (i.e., a regular and a lower threshold) for preserving general applicability of DCSPT to the habitat of different wireless transceivers that might exist in the network. In DCSPT, a wireless mesh node samples local channel busy time ($T_{\mathrm{busy}}^{\mathrm{local}}$) by passively monitoring local transmission activities at the level of *the regular CS-threshold*. This effective area from which $T_{\mathrm{busy}}^{\mathrm{local}}$ samples are taken is shown by the CSR_{C} ellipse in Fig. 2 for node C. By switching to *the lower CS-threshold*, a node can observe its extended surrounding area, to measure its CS-neighbors channel busy time ($T_{\mathrm{busy}}^{\mathrm{csn}}$). This effective area is shown by the $NCSR_{\mathrm{C}}$ ellipse in Fig. 2. Both these measurements are taken within an observation window of duration T_{p}. Selecting the appropriate *lower CS-threshold* level is important for the operation of DCSPT and this level could be derived from the node's *regular CS-threshold* by assuming an appropriate propagation model. Using $T_{\mathrm{busy}}^{\mathrm{local}}$ and $T_{\mathrm{busy}}^{\mathrm{csn}}$ observations, a node can calculate the percentage of channel utilization due to the transmissions that originate from the nodes outside its regular CS-range. Consequently, estimates of the channel utilization due to local transmissions, ρ_{local}, and the channel utilization due to transmissions at greater distance, ρ_{csn}, are given as follows:

$$\rho_{\mathrm{local}} = \frac{T_{\mathrm{busy}}^{\mathrm{local}}}{T_p}, \quad \text{and} \quad \rho_{\mathrm{csn}} = \frac{T_{\mathrm{busy}}^{\mathrm{csn}}}{T_p}. \tag{5}$$

Since ρ_{csn} already contains information ρ_{local} has, channel idle fraction for the purposes of AB estimation can thus be calculated as $\rho_{\mathrm{idle}} = 1 - \rho_{\mathrm{csn}}$. Extended observation area implied by the ρ_{csn} helps a flow-admitting node be more considerate in accepting a flow that might potentially violate some of its neighbors' QoS promises. However, note that using ρ_{csn} might also result in overly conservative AB estimates. For example, if node E streams to node G instead of node F in Fig. 2, the new flow at node C would not affect this transmission due to spatial reuse; still measuring ρ_{csn} unnecessarily accounts for it. More importantly, the self-interference-aware aggregate link utilization calculation method we outlined in Section 3.3 would also be similarly over-conservative. In particular, for the network in Fig. 1, node A being subject to interference from nodes in its CS-range (nodes N1, N2, N3, and N4) does not necessarily imply that some of this interference may not overlap in time. In other words, nodes N1 and N4 can simultaneously transmit. In general, this would reduce the effective number of contending nodes, N_{cont}, by one. Considering a linear topology incurring more than three hops, at least one pair of the flows can be transmitted simultaneously with an interferer considered by the dual CS approach, as we will discuss in the following paragraph.

Let us consider the linear network topology in Fig. 3 from the viewpoint of parallel transmissions and interferer locations. For the sake of brevity, we assume that nodes have a regular carrier sensing range of two times their transmission range. Let us consider the admitting node n, which is subject to flow self-interference from nodes $n-2$, $n-1$, $n+1$ and $n+2$. The closest possible interferer to node n (i.e., having the most detrimental effect), detectable by the dual-CS threshold approach, could be placed anywhere on node n's CS-range boundary. Transmissions from all possible locations on this boundary can only affect receptions of three nodes, in the worst case. Therefore, out of the six transmissions (including the one from the interferer) node n has to consider, at least two could take place simultaneously. For example, in Fig. 3, transmissions from the interferer node i can occur in parallel with the transmissions on both of the links $n-2 \rightarrow n-1$ and $n+1 \rightarrow n+2$. In Fig. 3(b) and (c), the interferer can transmit simultaneously only with one node: node $n-2$ and $n+2$, respectively. For the same linear topology in Fig. 3, having more than three hops, even without an interferer within extended CS-range, implies a pair of parallel transmissions. That is, transmissions on links $n-2 \rightarrow n-1$ and $n+1 \rightarrow n+2$ can happen simultaneously, reducing the effective N_{cont} by one in the calculation of ρ_{aggr}.

Consequently, the explained overlapping link utilization due to the parallel transmissions with the interferers outside the regular carrier sensing range can be calculated as

$$\rho_{\mathrm{overlap}} = \frac{T_{\mathrm{busy}}^{\mathrm{csn}} - T_{\mathrm{busy}}^{\mathrm{local}}}{T_p} \times R \cdot \left(\frac{L}{B} + T_{oh}\right), \tag{6}$$

where $R \cdot \left(\frac{L}{B} + T_{\mathrm{oh}}\right)$ represent the utilization requirement due to the transmission of the flow on one hop. Thus the link utilization due to the acceptance of the new flow, given in Eq. (4), should be updated, such that

$$\rho_{\mathrm{csn}}^{\mathrm{new}} = \rho_{\mathrm{csn}}^{\mathrm{old}} + \rho_{\mathrm{aggr}} - \rho_{\mathrm{overlap}}. \tag{7}$$

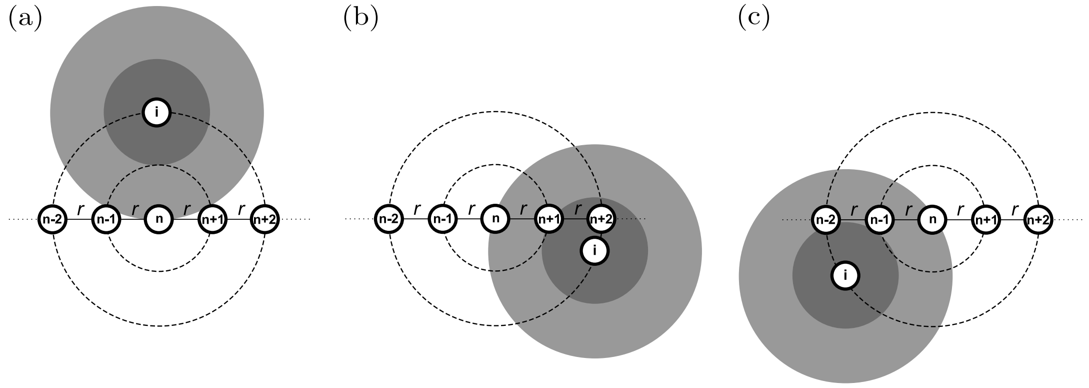


Fig. 3. Different positions for a worst-case hypothetical interferer, *i*, considered by the dual-CS approach on a linear topology. Regardless of its position, at least one hop simultaneous transmissions is always possible.

In Section 6, we quantify the benefits of using this parallel transmission awareness by carrying out ns-2 simulations of DCSPT method.

4.2. Packet probing with RTS/CTS handshake (PPRCH)

We have detailed DCSPT method for AB estimation in Section 4.1, however, an important issue needs to be addressed before we can use this method in today's mesh networks. Carrier sensing threshold adjustment capabilities are not readily offered by wireless transceiver vendors. Due to strict FCC requirements, vendors prefer closed radio designs and leave majority of the chipset functionalities only accessible to either a binary-distributed hardware abstraction layer software (HAL), such as Atheros-chipset based cards, or to the firmware that resides on the processor of the transceiver, such as Intel-chipset based cards. In both cases, dual-carrier sensing threshold scheme can not be implemented without explicit support from the vendor. We believe that a workaround is necessary for all near-future mesh deployments as well as the existing ones. Hence, we introduce the Packet Probing with RTS/CTS Handshake (PPRCH) method, which is practically feasible using off-the-shelf mesh networking equipment.

PPRCH incurs sending back-to-back probe packets, i.e., ordinary small packets that are only used for probing, and measuring their dispersion due to load on the channel. Gap between the packets are strongly correlated with the volume of the competing cross-traffic on the network [28]. Thus an algorithm can be used to derive the available bandwidth (AB) from the probe dispersion. The idea of packet probing has been mostly studied on wired networks [9], predominantly for end-to-end Internet bandwidth/capacity measurements. However, we are interested in possibility of using packet probing method for local available bandwidth estimation in a wireless mesh network. Most related work from this point of view is [14], where the authors described *AdHoc Probe* method exclusively to measure MAB for determining capacity of a wireless link, granted no background traffic exists. Our purpose here is to show that a proper packet probing variant is also suitable for estimating AB to support QoS routing in an operational wireless mesh network.

In our implementation, we measure the probe dispersion by taking the time difference between the completion times of two subsequent probe packet transmissions. Use of RTS/CTS is necessary (only for probe packets) to take advantage of IEEE 802.11 virtual carrier sense in capturing activities due to the hidden nodes around the measuring station. Sampling significant enough cross-traffic is the key to success when using PPRCH in a wireless mesh network. We designed and implemented PPRCH, as a two-module system within Linux kernel. This design, illustrated in Fig. 4, allows probes to sample local wireless traffic from mesh nodes.

The first module of PPRCH, *probe generator*, is responsible from generating the two probe packets, each transmitted from a different IEEE 802.11e [29] priority queue. This ensures the first probe packet is prioritized to mark the start of the AB measurement as soon as possible, and the second probe is sent out with a lower priority, favoring other mesh nodes to transmit their packets (i.e., because of the reduced probability of collision with the second probe). The second probe is always delayed by the amount dictated by the *gap adjuster* module. Hu and Steenkiste provides a detailed analysis on the interaction between the probe packets, probe gap, and the competing traffic in [9] and we adopt a variation of their PTR algorithm to use in our gap adjustment module, in the form given in Fig. 5. A mesh node finds the right AB measurement point by repeating the probes with re-adjusted gaps based on the observations on the dispersion of the previous probe. Convergence speed and accuracy of the gap search procedure can be adjusted by means of *delta* and *kappa* parameters of the algorithm. At the correct operating point, data rate used by the probes gives the AB. Care must be taken to consider the number of retransmitted probe frames (if any) when converting the dispersion to AB.

We have tested PPRCH method on the ORBIT testbed [31], which provides access to a 400-node wireless radio grid inside a well isolated experiment environment together with support services for experiment handling, measure-

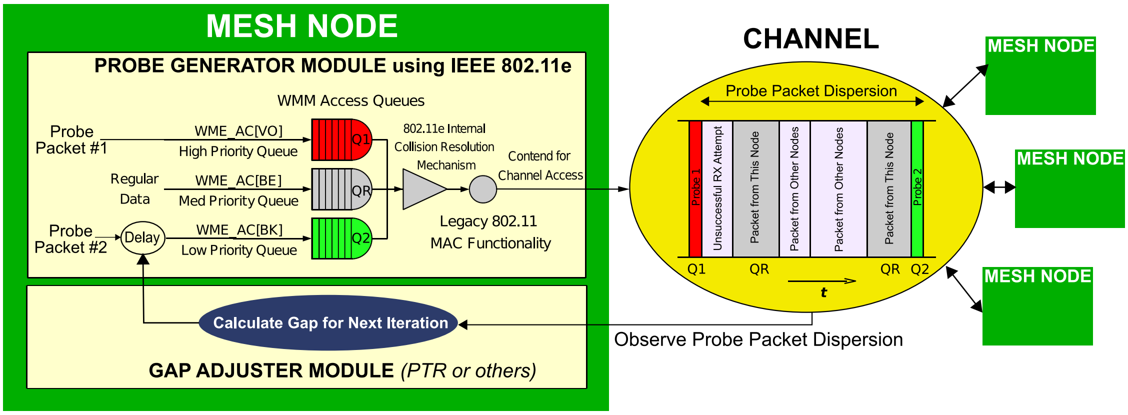


Fig. 4. Packet probing implementation has two functional modules on a mesh node. IEEE 802.11e priority queues work together with adjustable gaps for fast convergence to an AB estimate.

```
Accepts: PktTrainLength, PktProbeSize, LinkRate
Returns: AvailableBandwidth
InitialGap = PktProbeSize / (2 * LinkRate);
delta = InitialGap / 4;
kappa = 0.01;
repeat
    repeat
        send frame of PktProbeSize;
        wait InitialGap;
    until PktTrainLength frames sent;
    read ProbeDispersions[];
    err = [AVG(ProbeDispersions[]) – InitialGap] / InitialGap;
    InitialGap += delta;
until |err| < kappa;
return $\frac{PktProbeSize * (PktTrainLength-1)}{SUM (ProbeDispersions [])}$;
```

Fig. 5. A variant of PTR algorithm [9] is used to determine the right gaps, thus the AB from the dispersion of the packet probes.

ment collection etc. [32]. Some important features of the nodes, the software run on them, and the experiment parameters are summarized in Table 1. With the experiments presented here, we show that PPRCH can converge to dispersion values that can represent AB for each of the four different topologies given in Fig. 6. In the figure, arcs labeled with *T* indicate direction of the flow whose effect is to be measured. The arcs labeled with *P* arise from those hosts on which the measurements have been initiated. Each dispersion plot in Fig. 6 shows a consistent correlation between the cross traffic rate and the average probe dispersion, as we increased the cross traffic rate from 0 to 6 Mbits/s (since we used 802.11b at 11 Mbits/s). In all experiments, we have also observed that probe dispersion was not very sensitive to traffic rates less than 1 Mbits/s (or *utilization* < 18%), hence, care must be taken if more accurate estimation in that region is crucial. Nevertheless, precise AB measurements in real world applications are most needed in those channel utilization regions where probe dispersion shows strong correlation with the volume of traffic.

To compare the performance of PPRCH method for AB estimation in wireless mesh networks, we have implemented a conventional decodable-packets based AB estimation strategy on the ORBIT testbed, similar to those used in [5,10,11]. In Fig. 7(a) and (b), we present two AB estimation results from these experiments, using topologies (A) and (C) of Fig. 6 in a 1 Mbits/s IEEE 802.11b network. In topology (A), *node* 1 transmits a CBR flow of 200 Kbits/s by generating equally spaced UDP packets of size 1408-bytes to *node* 2. In topology (C), *node* 1 and *node* 4 are senders and can not decode each others packets (although they are within each others CS range). The common sink, *node* 2, can receive from both senders. Each sender transmits a 200 Kbits/s CBR flow similar to the topology (A). Flows were run between the 5th and 35th second of the experiment and the actual AB (*calculated offline*) is indicated with a dashed line in figures. We observe that decodable packet strategy performs well (less than 5% average Mean Square Error (MSE)), as long as all of the activity on air can be decoded as overheard packets (i.e., topology (A)). However, it fails to estimate actual AB in topology (C). As clearly seen in Fig. 7(b), accurate AB estimation was only possible at *node* 2 and the average MSE climbs to 33%. PPRCH[1] for the same two topologies, however, performs robust with less than 8.3% MSE in both cases.

Table 1
Attribute Summary for the ORBIT Experiments

Attribute	Value
Radio nodes	1 GHz VIA C3 Processor, 512 MB RAM, 20 GB HDD
Wireless interfaces	2X Atheros AR5212 based mini-PCI 802.11a/g
PHY/MAC used	IEEE 802.11b @ Channel 6
PHY link speed (fixed)	11 Mbits/s
Wireless output power	18 dBm
O/S used	Linux 2.6.12
Driver software	MadWifi r.1426 [30]

[1] For the PPRCH estimation, 4th order polynomial fit functions were used with the parameters given in Fig. 7(c) and (d). For MSE, mean residual error across all rates is reported.

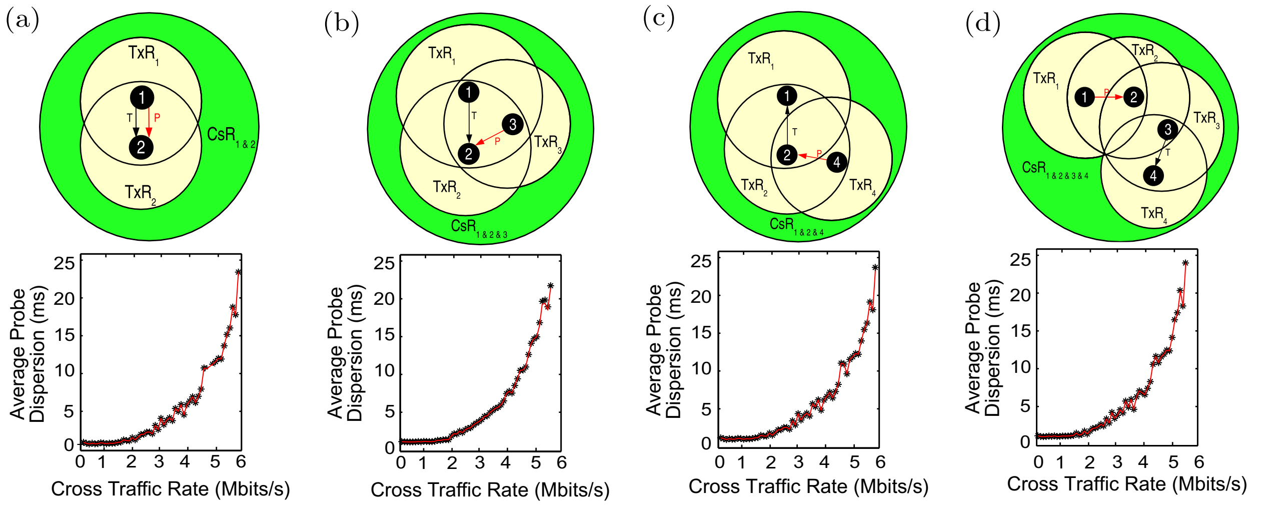


Fig. 6. PPRCH method being tested on real hardware for four different topologies (a)–(d) on the ORBIT grid. Corresponding average probe dispersions are plotted under each topology.

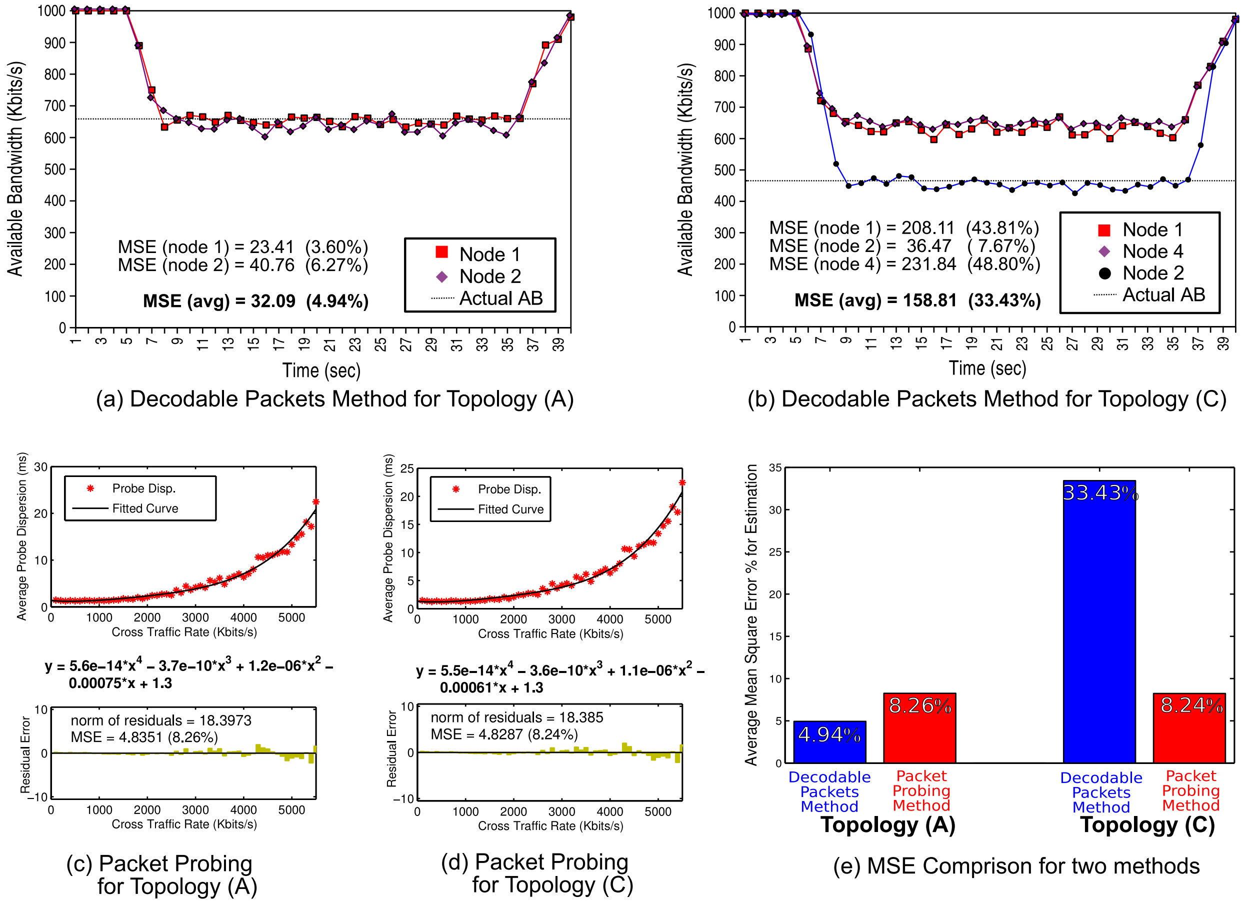


Fig. 7. AB estimation performance comparison experiments. Decodable-packets method and packet probing is compared for topologies (a) and (c) of Fig. 6.

Comparison of the average MSE for both methods are summarized in Fig. 7(e). These results motivate the use of interference-aware approaches, such as PPRCH, for the estimation of AB in wireless mesh networks.

5. Admission control and routing protocol integration

In Section 4, we have covered the methods to realize AB estimation. In this section, we will first discuss our distributed routing and resource control protocol which allows mesh nodes to use AB estimates for controlling flow admission to the network. Our protocol design enhances a lightweight hop-by-hop routing protocol implementation, LUNAR [33], by integrating AB estimation and admission control methods described in previous section.

5.1. Admission control decision

We have shown how link utilization would change with the acceptance of a new flow in Eq. (7). To avoid congestion, the admitting node must ensure $\rho_{\text{csn}}^{\text{new}} \leqslant 1$. For more conservative admission control, the utilization limit can be chosen smaller than *one*, allowing some tolerance to estimation errors. We will use a limit value of *one* here, for the sake of simplicity. Consequently, when a node makes an admission decision, it needs to check if

$$\rho_{\text{csn}}^{\text{new}} = \rho_{\text{csn}}^{\text{old}} + \rho_{\text{aggr}} - \rho_{\text{overlap}} \leqslant 1 \tag{8}$$

can be satisfied. This condition can be re-written using Eqs. (5) and (6)

$$\rho_{\text{aggr}} \leqslant 1 - \rho_{\text{csn}}^{\text{old}} + \rho_{\text{overlap}} \tag{9}$$

$$\rho_{\text{aggr}} \leqslant 1 - \frac{T_{\text{busy}}^{\text{csn}}}{T_p} + \frac{T_{\text{busy}}^{\text{csn}} - T_{\text{busy}}^{\text{local}}}{T_p} \times R \cdot \left(T_{\text{oh}} + \frac{L}{B_a}\right), \tag{10}$$

where the left side represents the aggregate link utilization required by the new flow, and the right side represents the fraction of the channel available to the flow. In a network where all the links use the same rate, Eq. (10) can be simplified to:

$$(N_{\text{cont}} + 1)\rho_{\text{req}} \leqslant 1 - \frac{T_{\text{busy}}^{\text{csn}}}{T_p} + \frac{T_{\text{busy}}^{\text{csn}} - T_{\text{busy}}^{\text{local}}}{T_p} \times \rho_{\text{req}}, \tag{11}$$

where ρ_{req} was given in Eq. (3). This condition must be verified at each node during the route establishment phase and we provide details of this process in the following section.

5.2. Protocol design

We have chosen Lightweight Underlay Network Ad-Hoc Routing (LUNAR) [33] protocol for integrating our DCSPT-based admission control scheme for wireless mesh networks. Our decision on LUNAR is based on its simplicity and small code base. The concepts under discussion, however, can also be applied to other hop-by-hop routing protocols such as AODV. The implementation can be used with a network that supports arbitrary number of rates.

To keep track of flow information, a *Flow Table* is maintained at every node in the network. Each entry of this table describes a flow passing through the node, including the *source*, *destination*, *amount of reserved bandwidth*, and *status*. All entries are maintained as soft state, hence they are automatically deleted if the topology changes or flows time out. The status can be one of the three values: *requesting*, *reserved*, and *activated*. To support multiple flows between the same source-destination pair, an additional unique flow identifier, *Flow ID* indexes the *Flow Table*. Structure of this table is illustrated in Fig. 8.

The extra information fields added to the routing protocol help distributing topology and flow requirement information to each node, so that nodes can make an admission decision. Upon receiving an ARP request, the source node broadcasts a route request (RREQ) to its neighbors. Besides the original fields defined in LUNAR, we add four more fields to the RREQ message to facilitate QoS provisioning. "*Flow ID*", associated with the source and destination, uniquely identifies a flow. "*Sending Rate*" specifies average packet sending rate R of the flow. We also include the packet size "L" in the RREQ. L and R characterize the load offered by the flow. "B_{upstream}" field specifies the link rate used by the previous hop, which is 0 at the source node since it is the first hop. B_{upstream} is used by the next-hop node to calculate the aggregate link utilization required by the requesting flow.

The operation of our integrated routing and resource management protocol is detailed in the following paragraphs. A simplified version of the described algorithmic flow is also illustrated in Fig. 9, for the convenience of the reader. Upon receiving the RREQ, each intermediate node calculates the aggregate link utilization introduced by the transmissions over last the two hops (recall the N_{cont} approximation discussion in Section 3.2) and the next hop. Link utilization of the immediate previous hop is calculated as $R \cdot \left(T_{\text{oh}} + \frac{L}{B_{\text{up}}}\right)$, where B_{up} denotes the link rate. Link utilization of the hop before the immediate previous hop is $R \cdot \left(T_{\text{oh}} + \frac{L}{B_{\text{upstream}}}\right)$, where B_{upstream} is obtained from the received RREQ. Adding its own transmission on the next hop, the node can calculate the aggregate channel utilization as $\rho_{\text{aggr}}^{\text{up}} = R \cdot \left(T_{\text{oh}} + \frac{L}{B_{\text{up}}}\right) + R \cdot \left(T_{\text{oh}} + \frac{L}{B_{\text{upstream}}}\right) + R \cdot \left(T_{\text{oh}} + \frac{L}{B_{\text{down}}}\right)$, where B_{down} is the link rate node will use for the next hop. This aggregate link utilization of the flow is checked against Eq. (10), and the admission decision is made accordingly. If the requirement can be met, then the

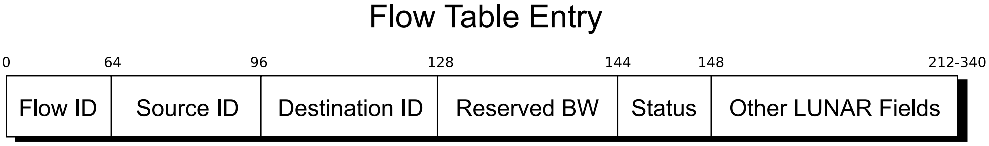


Fig. 8. Structure of the flow table used in protocol implementation.

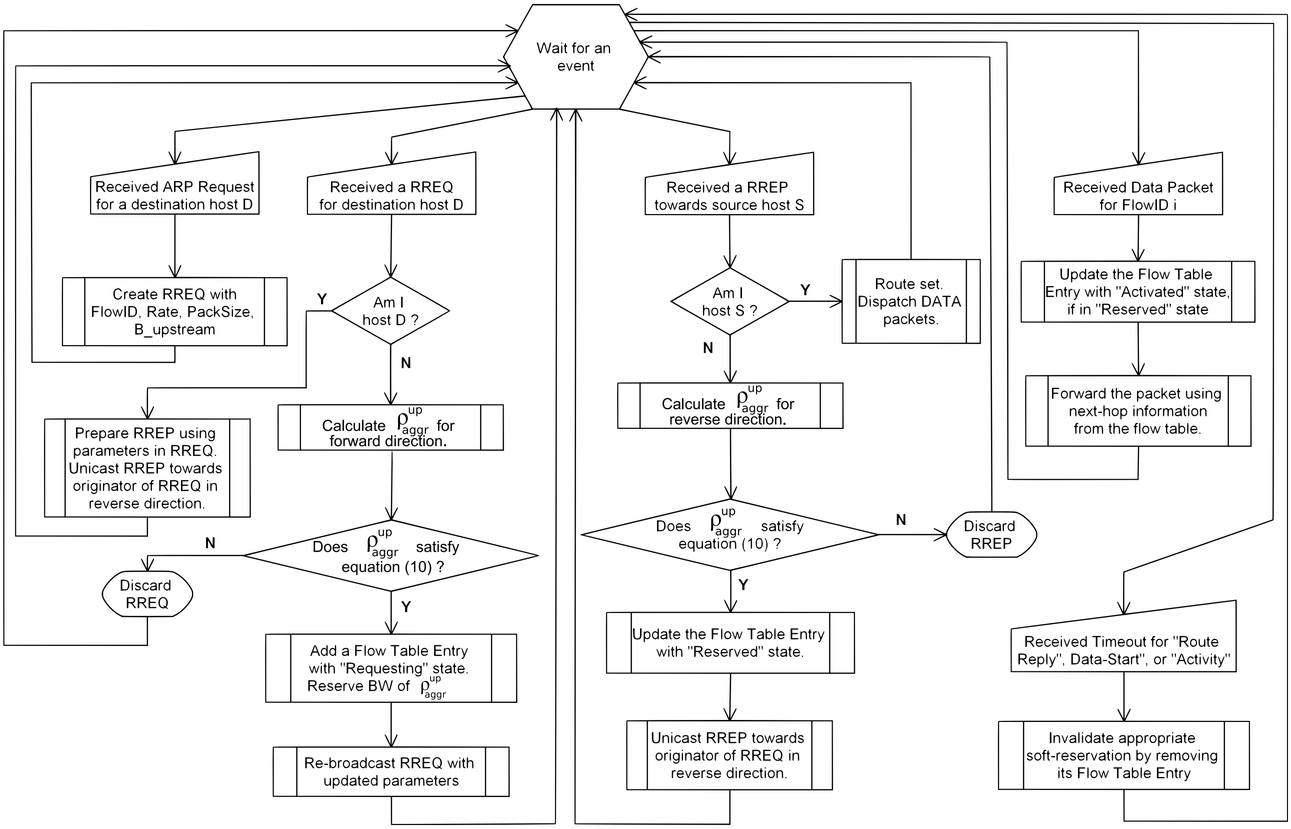

Wait for an event
Received ARP Request for a destination host D
Create RREQ with FlowID, Rate, PackSize, B_upstream
Received a RREQ for destination host D
Am I host D ?
Y
N
Prepare RREP using parameters in RREQ. Unicast RREP towards originator of RREQ in reverse direction.
Calculate ρ_{aggr}^{up} for forward direction.
Does ρ_{aggr}^{up} satisfy equation (10) ?
N
Y
Discard RREQ
Add a Flow Table Entry with "Requesting" state. Reserve BW of ρ_{aggr}^{up}
Re-broadcast RREQ with updated parameters
Received a RREP towards source host S
Am I host S ?
Y
N
Route set. Dispatch DATA packets.
Calculate ρ_{aggr}^{up} for reverse direction.
Does ρ_{aggr}^{up} satisfy equation (10) ?
N
Y
Discard RREP
Update the Flow Table Entry with "Reserved" state.
Unicast RREP towards originator of RREQ in reverse direction.
Received Data Packet for FlowID i
Update the Flow Table Entry with "Activated" state, if in "Reserved" state
Forward the packet using next-hop information from the flow table.
Received Timeout for "Route Reply", Data-Start", or "Activity"
Invalidate appropriate soft-reservation by removing its Flow Table Entry

Fig. 9. Flowchart describing the operation of the integrated routing and resource reservation protocol.

node forwards (i.e., re-broadcasts) the RREQ message. Also it inserts an entry into its *Flow Table*, recording source, destination and ID of the flow, filling in the "*reserved bandwidth*" field with $\rho_{\text{aggr}}^{\text{up}}$ and setting the status to "*Requesting*". If the requirement can not be satisfied, then the node silently discards the RREQ. Admission control performed in this forward phase is *partial*, because the admitting node can only obtain the information of its upstream contending nodes. Although it under-estimates the aggregate channel utilization introduced by the flow, it is useful as the first pass to weed out certain non-qualified routes.

Upon receiving the route request, the intended destination sends back a route reply (RREP) if the AB is large enough. Note that multiple copies of RREQ might arrive along different paths to the destination. To increase the possibility of discovering a qualifying path, the destination sends back a RREP for each copy of the RREQ. We add the same four fields in RREQ to RREP and at each forwarding node, the *full admission control* is performed. It is called *full* since in this reverse phase, the total link utilization required by the flow can now be calculated completely. Each forwarding node calculates the aggregate utilization over the previous two hops the same way as in RREQ phase. Note that the *preceding* hops in the forward direction are *succeeding* hops in the reverse direction. Consequently, by summing up the aggregate link utilizations calculated during the RREQ and RREP phases, an admitting node can get updated total channel utilization ρ_{aggr} of the requesting flow. Again ρ_{aggr} is tested against Eq. (10) to make the final admission decision. If satisfied, RREP is forwarded to the next hop and this soft-state reservation is reflected to the *Flow Table* as "*Reserved*".

It is only after the appearance of the flow's first data packet, nodes update the status of the corresponding entry to "*Activated*", implying that the indicated bandwidth is actually in use. When a channel reservation becomes "*Reserved*", $\rho_{\text{resv}} = \rho_{\text{aggr}}$ is no longer available to other flows and thus the idle fraction of the channel becomes $1 - \rho_{\text{local}} - \rho_{\text{resv}}$.

Although nodes make reservations during the RREP phase, the route may not be successfully established due to a variety of reasons, such as link failures or the source deciding to use a different route [6]. Therefore, reservations must be maintained as soft-state and are deleted when the associated timer expires. Specifically, each entry is associated with three timeout values: *route-reply timeout*, *data-start timeout*, and *activity timeout*. The *route reply timeout* is used to remove a entry from the *Flow Table* if a node only receives a route request, but does not receive the corresponding route reply in time. The *data start timeout* expires if a node has received a route reply for a reservation, but has not yet received any data frame from the source within this period. The *activity timeout* is used to keep track of flows which have not transmitted any data frame for a long period of time, possibly due to mobility incurred route failures.

5.3. QoS violation and recovery

We exploit the periodic route refreshment mechanism used in LUNAR to provide an implicit QoS violation detection and recovery mechanism. If a node can no longer meet its QoS commitments made to existing flows, the admission control mechanism will reject or find alternate routes for some of these flows during the next route refreshment (by default every 3 s) attempt. Hence, when a new route request is received, a node should first delete any existing reservation with the same flow ID, as to prevent double counting of the bandwidth requested by this flow.

Refreshing path reservations creates a challenge in providing path stability. A flow should switch paths only when the current path can no longer meet the flow QoS requirement. Unnecessary changes may lead to changes to other existing flows, likely resulting in some flows alternating between an admitted and rejected state. This phenomenon is closely related to the way source nodes decide which of multiple possible routes to use (recall that a source node may receive multiple route replies). To increase stability, we add an *On Current Path* (OCP) flag in the RREQ and RREP packets, and force the source nodes to reuse a current path if available. Specifically, when the source node initiates route refreshment, it sets the OCP flag. When an intermediate node that is not on the current path receives RREQ, it resets the flag; otherwise, it leaves the flag unchanged. The destination copies the flag from the RREQ into the RREP packet. Upon receiving RREPs, the source node will favor the path (if any) with OCP flag set over the others.

5.4. Scalability of the protocol

The most important barrier in front of the wide-spread adoption of many QoS-aware routing and reservation management techniques has been the scalability issues due to the burden of per-flow state management at the routers. A well-known example to this is the IntServ [16] protocol, proposed for the core networks of the ISPs. Fortunately, the scale of the storage and processing requirements imposed by per-flow state maintenance for these core networks (like Internet) is very different than the edge networks (like Wireless Mesh Networks). Typical wireless link capacities, average-hops-to-infrastructure, and the number of nodes-per-square-foot place a fair restriction on the number of flows a given mesh gateway has to handle, in contrast to a core router of a typical ISP. Considering today's state of the art routers (with over 256 K state entries) and the Moore's law for CPU and memory ICs, we envision that it is unlikely that wireless mesh routers operating at the edge network will face scalability problems due to per-flow state maintenance. Also, multi-radio/multi-channel mesh networks supporting flow-partitioning (i.e. resource management only for particular flows on designated channels) would further alleviate potential scalability issues for very small and cheap routers with limited memory and processor resources.

The protocol that constitutes the base for our implementation, LUNAR, on the other hand, has a design which eliminated the necessity for many complex control-plane operations, such as explicit route caching, route maintenance and salvation [33]. This has been another motivation towards a scalable deployment of our proposal for wireless mesh networks.

6. Performance analysis

In this section, we evaluate the performance of our integrated routing and admission control mechanism by simulating it over a fixed-grid (on a 1500 m × 1500 m terrain) and several random (on a 1000 m × 1000 m terrain) wireless mesh network topologies. For available bandwidth estimation, we use the DCSPT method described in Section 4.1. All simulations are conducted using the ns-2 version 2.29 with default parameters [24]; nodes have a transmission range of 250 m and a CS-range of 550 m. Each simulation experiment lasts at least 1.5 min, and several runs with different seeds were executed before commenting on the outcomes. Constant bit-rate UDP flows (1500-byte packets) with varying rates (as detailed in the following subsections) are used to simulate the workload of the mesh network. The IEEE 802.11 is used as the MAC layer protocol with a fixed link rate of 2 Mbits/s, unless noted otherwise. We have implemented our admission control scheme by extending the LUNAR implementation version 20041221 from Uppsala University [34]. QoS is evaluated using cumulative system throughput and its variance in per-flow measurements, as well as per-flow delay and its variance. The performance of the AB estimation mechanism can be judged by the number and size of flows admitted while maintaining a good QoS-level for the mesh nodes.

In the following subsections, we summarize representative results from three different groups of simulation experiments. First one is carried out to verify operation of AB estimation and admission control under a controlled/tractable workload scenario. The second experiment aims to shed light into the performance observed by the mesh nodes of a random topology with heavier workloads. Finally, we investigate various random topologies to justify benefits of parallel transmission awareness in our admission control mechanism.

6.1. Verification of effective operation

The purpose of this experiment is to demonstrate the effectiveness of our integrated QoS-aware routing mechanism, by inspecting its operation on a tractable wireless mesh topology. For this reason, we consider the 5-by-7 grid network topology consisting of 35 nodes, as shown in Fig. 10. The inter-node distances are 250 meters in N–S and E–W directions.

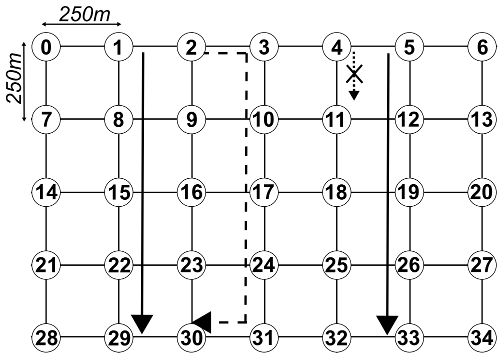
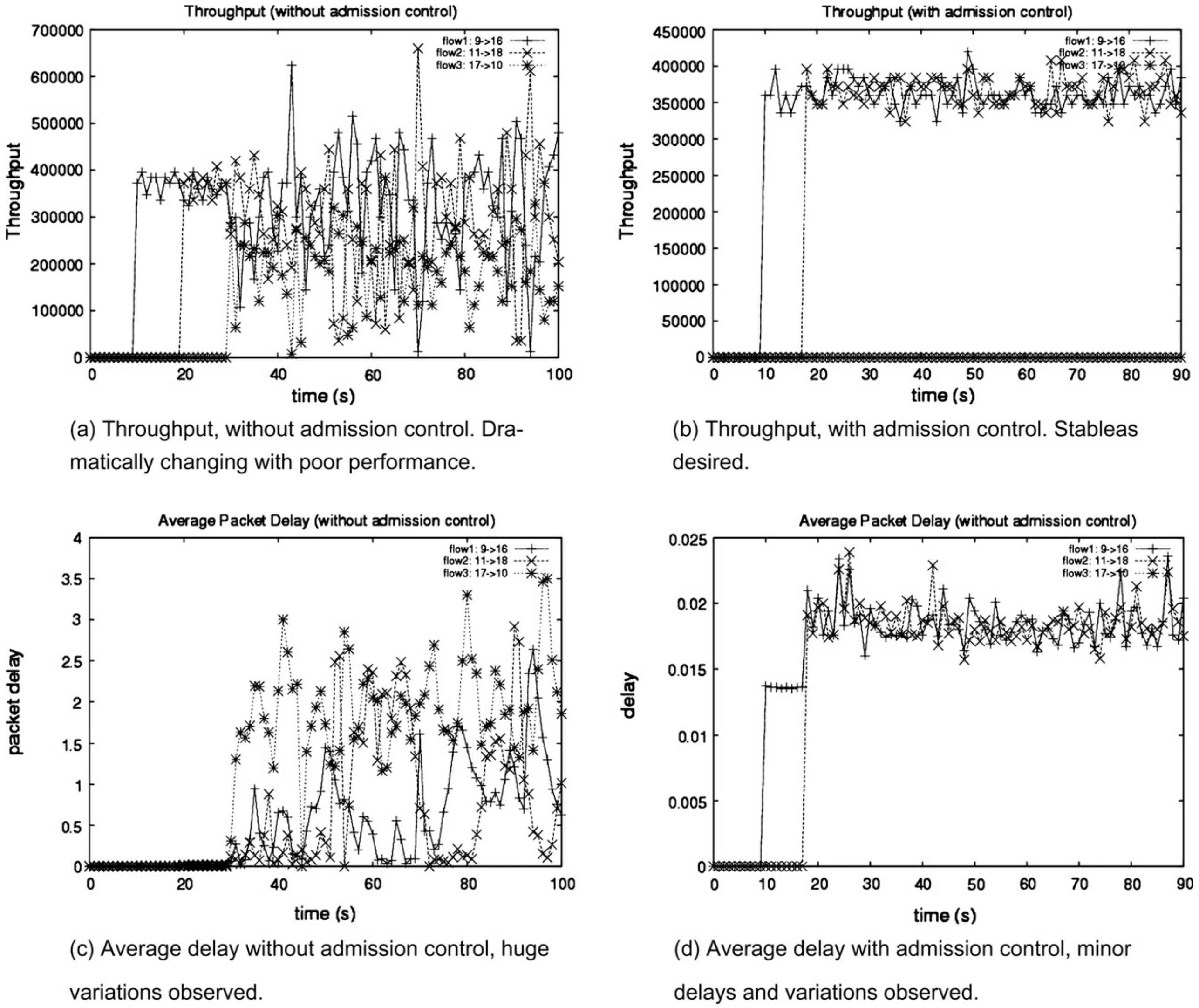


Fig. 10. The first simulated network topology: *A grid.*

In the experiment, we test if the routing mechanism can find a longer path, when the shortest one can not meet the link utilization requirements of the flow. Similarly, if no such suitable paths exist in the network, would our scheme reject the flow gracefully? For this scenario, node 1 starts with a flow destined to node 29 and soon after, node 5 starts the second flow to node 33. Both flows carry packets from a data source of 256 Kbits/s rate. Hence, each flow implies a link utilization of 0.128 on a single isolated link. Since the max. ρ_{aggr} calculated for each of the flows is $4 \times 0.128 = 0.512$ on their shortest paths, they are admitted along their respective shortest paths, as shown in Fig. 10. Few seconds later, node 2 attempts to start a flow to node 30, again with a bandwidth requirement of 256 Kbits/s. During the route establishment phase, node 16 is able to account for all four transmissions that are occurring, through its dual-CS test, and stops participating in forwarding route requests any further because of the lack of AB visible to it. In the simulation, as shown by the dashed arrow in Fig. 10, route establishment succeeds



(a) Throughput, without admission control. Dramatically changing with poor performance.

(b) Throughput, with admission control. Stableas desired.

(c) Average delay without admission control, huge variations observed.

(d) Average delay with admission control, minor delays and variations observed.

Fig. 11. Quantifying benefits of admission control in terms of throughput and average end-to-end delay. (a) Throughput, without admission control. Dramatically changing with poor performance. (b) Throughput, with admission control. Stable as desired. (c) Average delay without admission control, huge variations observed. (d) Average delay with admission control, minor delays and variations observed.

over the relatively cleaner path, through the nodes 3, 10, 17, 24, and 31. Finally, another flow of 256 Kbits/s is started from node 4 to 32. Since none of the potential paths can guarantee sufficient bandwidth to support this flow, it is denied admission and times out due to not being able to find a route to the destination node 32.

6.2. Quantifying QoS enhancement through admission control

The purpose of the experiments in this section is to quantify the QoS improvement that our integrated routing and admission control mechanism provides for the flows in the network. We first take a close look at the following scenario tested on the topology given in Fig. 10. In this scenario, nodes use 1 Mbits/s fixed link rates, and three flows, between nodes 9 and 16 starting at 10th s, between nodes 11 and 18 starting at 20th s, between nodes 10 and 17 starting at 30th s, are considered. Applications generating these flows require 380 Kbits/s from the network. Per-flow throughput and average packet delay values are plotted with and without our admission control scheme in Fig. 11. It is clearly visible from both Fig. 11(a) and (c) that acceptance of the third flow into the network causes immediate congestion. Once the channel becomes congested, the delays for all the three flows increase greatly, along with large variations. Such a high delay with unstable throughput is often fatal for most real-time applications. When the admission control is enabled, third flow can not be admitted anymore due to the lack of enough channel capacity, resulting in the stable throughput performance in Fig. 11(b) with average delays at around 1/100 of the values observed when AC is not used, as plotted in Fig. 11(d). The short packet delays and consistent throughput demonstrate that our scheme can be used to support real-time applications, such as video and multimedia over wireless mesh networks.

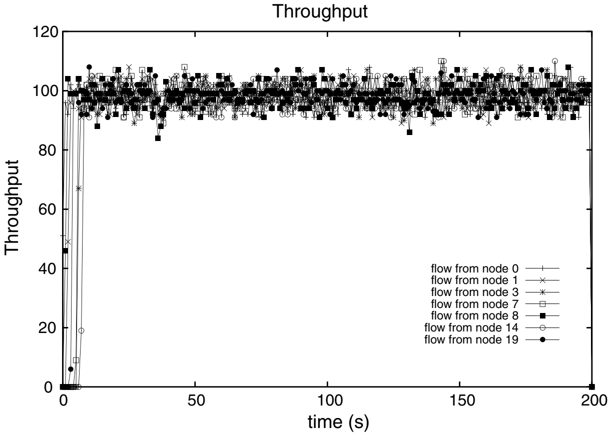
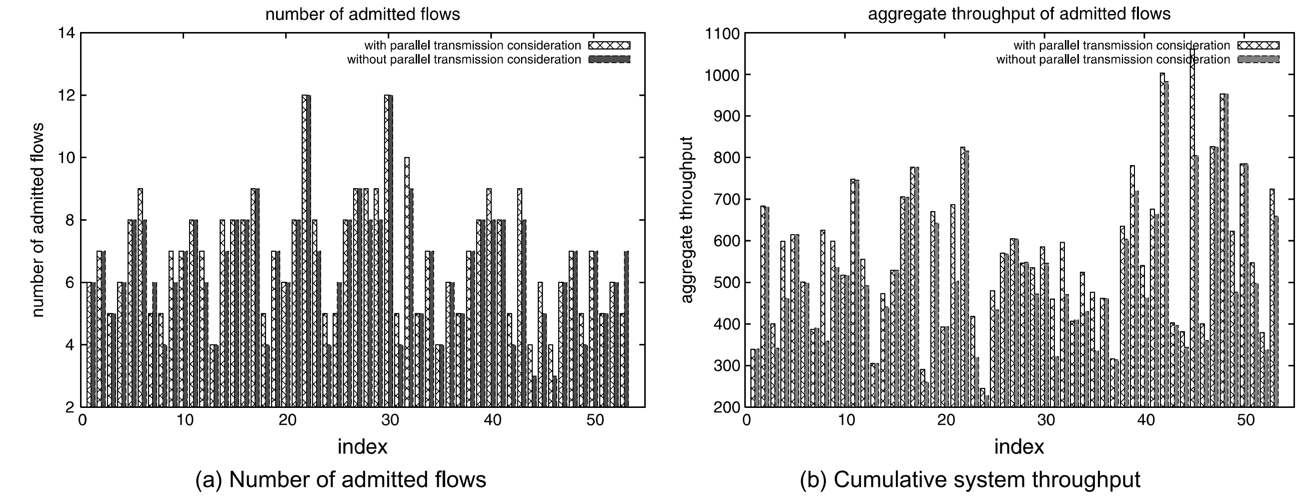


Fig. 12. In the 20-node scenario, only 7 flows are admitted, all achieving a stable throughput performance.

Next, we look at a larger scenario on a 1000 m × 1000 m terrain with 20 randomly positioned mesh nodes. Each node attempts to establish a 100 Kbits/s flow towards a randomly chosen destination at a random time within the first 20 s of the experiment. This configuration, also used in the remaining experiments, has a maximum forwarding path of 1414 m, a network diameter of 5.66 hops with nodes having 3 to 4 neighbors on average, as calculated by SCORES [35]. Throughput performance from a sample run of this experiment with admission control enabled is given in Fig. 12. Results from the other runs were similar to the one presented here. Each vertical line in the figure represents the acceptance of a new flow, so we can see that only seven flows have been admitted. However, the admitted flows achieved a constant throughput around 100 Kbits/s with low delay variance, matching the application QoS requirements.



(a) Number of admitted flows

(b) Cumulative system throughput

Fig. 13. In 19 of the networks tested, more flows could be admitted with parallel transmission awareness; including those, a total of 31 experiments showed improvements in cumulative system throughput.

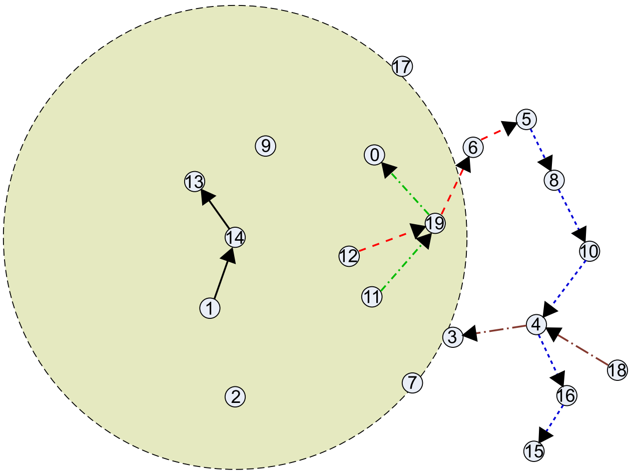
6.3. Advantages of parallel transmission awareness

The following experiments demonstrate the benefits of parallel transmission-awareness of our DCSPT method, as outlined in Section 4.1. Note that in our simulations, we have only implemented and tested the calculations that consider simultaneous transmission opportunities between a given flow and the interferers forwarding other flows. Therefore, it would not be unrealistic to expect better performance upon introducing inter-flow parallel transmission awareness into the experiment. Next release of our implementation is planned to have the support for intra-flow parallel transmission awareness.

We use the same 1000 m × 1000 m scenario with 20 randomly positioned nodes. This time however, to increase the variety of tested workload patterns, each random mesh node picks a random flow rate between 0 and 300 Kbits/s destined towards a randomly selected mesh node. We simulate 53 instantiations of this experiment, each with a different topology and workload. In Fig. 13(a), we compare total number of admitted flows for each of the runs with and without parallel transmission awareness. Fig. 13(b) illustrates the change in throughput for the corresponding topologies. We have observed that parallel transmission awareness increased number of admitted flows in 35.8% of the topologies, while keeping the same number of flows in 58.6%. In some marginal cases (i.e. remaining 5.6%), number of accepted flows decreased due to parallel transmission awareness. This can be attributed to the changed bandwidth consumption pattern due to the acceptance of some flows that would not have been admitted without parallel transmission awareness.

We observed that 58.5% of the simulated networks experienced throughput performance improvements, while remaining 41.5% have not observed any change. This indicates that parallel transmission-awareness brings throughput increase to not only those networks in which more flows were accepted, but also to the others retaining the same accepted number of flows. Having a higher number of alternate routes, is the major reason behind the better overall network utilization. The total cumulative system throughput improvements of up to 80% were observed among the topologies tested. (i.e., for the network with index #31). It is also worth noting that the performance of our method without parallel-transmission awareness is comparable to that of the method presented in [3].

To explain parallel transmission related performance improvements better, we illustrate the relevant part of the network topology #8 together with the flow details in Fig. 14. From Fig. 13(a), we observe that parallel transmission-awareness increased number of accepted flows from *four* to *five* in this topology. The flow shown with ID 5 is this additionally accepted flow into the network. Let us focus on the bandwidth availability as seen by the bottleneck node 14, whose CS-range is denoted by the dashed circle. After the flows with ID 1,2,3, and 4 are admitted, node 14 sees a local utilization of, $\rho_{\text{local}} = \frac{(2\ \times 48\ \text{Kbps}) + (2 \times 22\ \text{Kbps})}{2000\ \text{Kbps}} = 0.07$, using the observation via its regular CS-threshold. At the same time, using its lowered CS-threshold, node 14 observes the utilization available to it as $(1 - \rho_{\text{csn}}) = 1 - (\rho_{\text{local}} + (1\ \times\ 48\ \text{Kbps}) + \frac{(5 \times 185\ \text{Kbps}) + (2\times 170\ \text{Kbps})}{2000\ \text{Kbps})=0.2735}$. Since the fifth flow under consideration needs an additional utilization of $\frac{(2\times 285\ \text{Kbps})}{2000\text{Kbps}} = 0.285$ from node 14, the Eq. (8) can not be satisfied, and the flow should be rejected if there is no parallel transmission-awareness in the network. However, as transmissions from this new flow can occur simultaneously with transmissions from many other flows in the network, the fifth flow does not create congestion anywhere in the network. Thus, it is accepted by our admission control scheme, increasing cumulative system throughput by an additional 570 Kbps of traffic. Yet, we have also observed that such throughput gains did not compromise the QoS of admitted flows for the topologies we have simulated. In Fig. 15, we plot throughput improvements exhibited by the 31 different topologies versus their packet



FLOWS IN TOPOLOGY #8

ID	Source	Destination	Requested
1	11	0	22Kbps
2	12	5	48Kbps
3	5	15	185Kbps
4	18	3	170Kbps
5	1	13	285Kbps

Fig. 14. Looking at network topology #8 in detail. Parallel transmission-awareness allows flow 1 → 14 → 13 to be admitted, increasing cumulative system throughput by 570 Kbits/s.

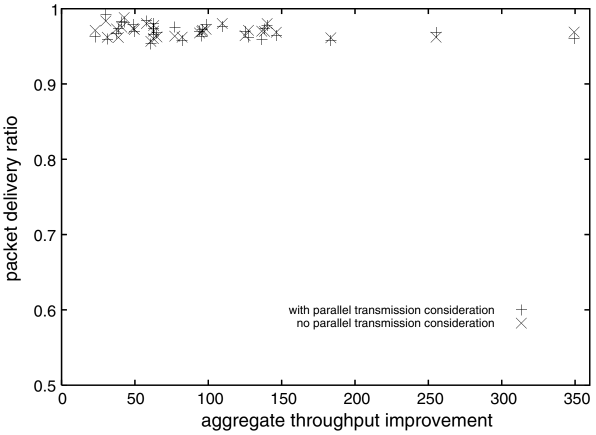
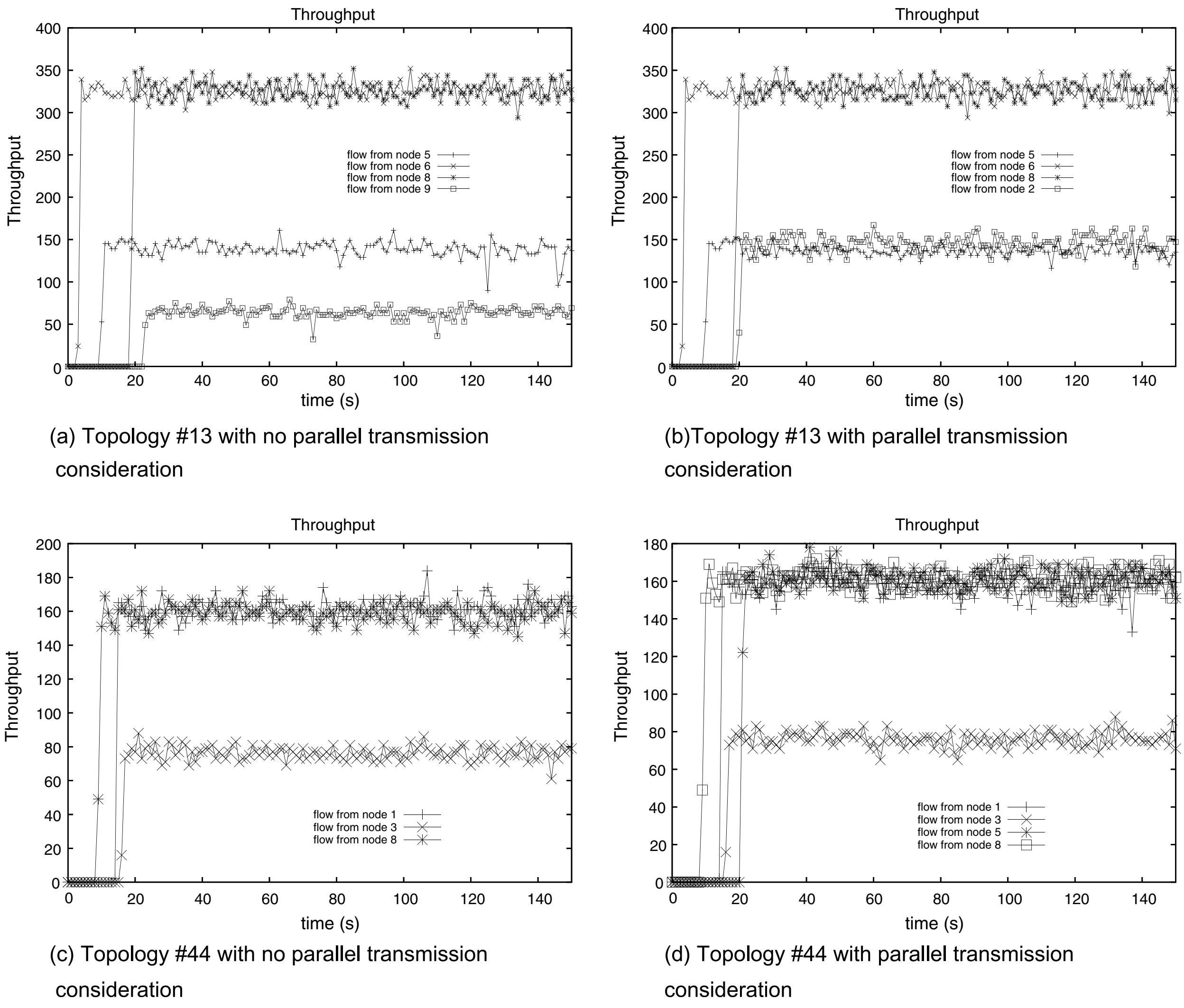


Fig. 15. Cumulative system throughput improvements with parallel transmission awareness have no significant effect on PDR; in most experiments, 50–150 Kbps more traffic could be admitted.

delivery ratios (PDR). It is clear that throughput gains come without a significant cost in terms of PDR performance. In most experiments, parallel transmission-awareness allowed an additional traffic of 50–150 Kbps, demonstrating the importance of better AB estimations in wireless mesh networks.

Fig. 16 illustrates throughput performance results from two topologies: #13 and #44. In the former (Fig. 16(a) and (b)), parallel transmission awareness do not affect the number of flows admitted. However, instead of the flow from node 9, it allows the earlier flow from node 2 to be admitted into the network, which had a much higher bandwidth requirement. In the latter, one more flow could be admitted when parallel transmissions were considered, increasing cumulative system throughput by 160 Kbps. In both cases, there is no evident increase in throughput variance, despite the increased system utilization.



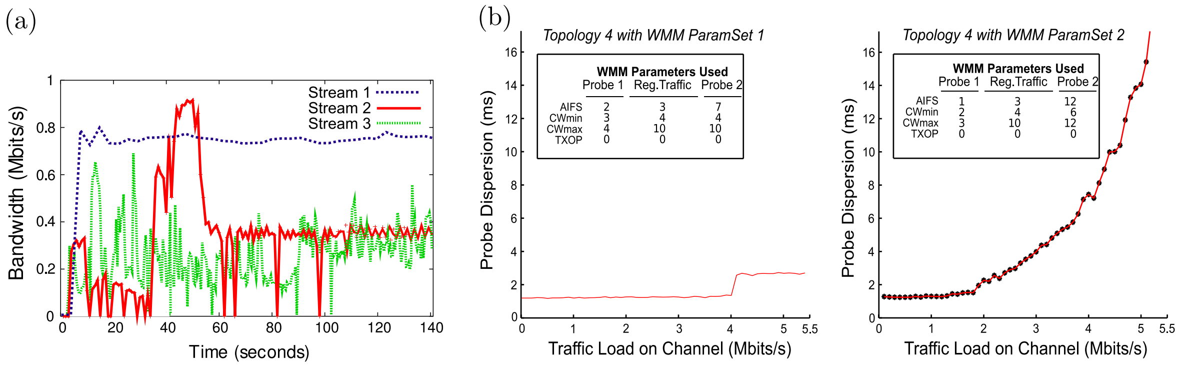
(a) Topology #13 with no parallel transmission consideration

(b)Topology #13 with parallel transmission consideration

(c) Topology #44 with no parallel transmission consideration

(d) Topology #44 with parallel transmission consideration

Fig. 16. Quantifying gains from parallel transmission-awareness. In topology #13 (a and b), the same number of flows are admitted, but not considering parallel transmissions causes rejection of a bigger flow. In topology #44 (c and d), one more flow is admitted, increasing cumulative system throughput.



WMM Parameters Used (ParamSet 1)

	Probe 1	Reg.Traffic	Probe 2
AIFS	2	3	7
CWmin	3	4	4
CWmax	4	10	10
TXOP	0	0	0

WMM Parameters Used (ParamSet 2)

	Probe 1	Reg.Traffic	Probe 2
AIFS	1	3	12
CWmin	2	4	6
CWmax	3	10	12
TXOP	0	0	0

Fig. 17. Experiences from our prototype implementation: (a) bandwidth requirements of three different video streams used in evaluations; each has very different characteristics. (b) IEEE 802.11e parameter values (WMM) affect probe dispersion behavior.

7. Experiences from prototype implementation and discussion

We have implemented and successfully demonstrated a complete system prototype for wireless video streaming on the ORBIT testbed, using the schemes described in this paper. In this section, we briefly discuss some of the observations we had while designing and testing the system, since we believe that these issues are also relevant to other wireless mesh networks planning to use schemes similar to the ones described in this study.

7.1. Stability of workload characteristics

In our system demonstrations, we have used small video clips (about two-minutes long) from commercial movie trailers. We have experienced that for some video streams, specifying average inter-packet times, ($\frac{1}{R}$, as explained in Section 3.1), were harder than others. Fig. 17(a) illustrates bandwidth requirements for three of the streams we have used in our tests. It is clear that, depending on the used video encoding method and the degree of action in the clip, a stream might exhibit a long-term constant $\frac{1}{R}$ (e.g. Stream 1), a short-term constant $\frac{1}{R}$ (e.g. Stream 2), or a highly-varying $\frac{1}{R}$ (e.g. Stream 3) behavior. This makes *AB sampling interval*, *AB smoothing filters* (i.e., linear or non-linear), and *route refreshment periods* very important system parameters that administrators should not overlook. Dynamic adjustment of such parameters for optimal performance remains an open research question. For our prototype implementation, we have sampled AB every second and fed measurements to a moving average smoothing filter with a strong low-pass characteristic (i.e., history weight $\alpha = 0.8$).

7.2. Packet probing success

As we discussed in Section 4.2, we used packet probing as a practical AB estimation method that can be implemented using off-the-shelf IEEE 802.11 wireless cards. We have observed two important points that are critical to the success of PPRCH method. Probing implementations should be careful about the way dispersion measurements are taken. The methodology should be as accurate as possible while keeping it non-intrusive to the fullest extend. We found that recording the times between the network device interrupts at kernel-space, using the CPU Time Stamp Counter (available on Pentium-compatible CPUs) as the most appropriate method for our setup. For the proper operation of this method, the network device driver feature that aggregates per-packet interrupts into bunches must be disabled. System designers should also exercise caution when selecting IEEE 802.11e priority queue parameters, used by the probe packets. In some experiments, we have seen that insufficient prioritization resulted in very slow (or no) convergence to right dispersion values. An example observation is given in Fig. 17(b), where two sets of priority queue parameters result in very different dispersion readings for the same network.

8. Conclusion

We have presented an admission control mechanism for IEEE 802.11-based wireless mesh networks (WMNs), integrated with a routing protocol. Our available bandwidth-aware routing mechanism discovers routes that can satisfy flow rate requirements without affecting ongoing flows in the network. The admission control algorithm is also able to calculate available bandwidth and predict the bandwidth consumption of a flow accurately while taking into account parallel transmissions. Alternatively, a packet probing-based available bandwidth estimation method, which can be implemented even on today's WMN hardware, is introduced and tested. Simulation results show that integrating admission control into the routing protocol enables the identification of alternate routes if the shortest path is congested. Admission control can limit the amount of data traffic in the network to provide QoS guarantees to admitted flows. By exploiting channel reuse, our scheme can admit more traffic while maintaining QoS, compared to mechanisms that do not account for parallel transmissions. Through parallel transmission awareness, cumulative system throughput improvements of up to 80% were observed in our experiments. We have also shared some insights gained from our prototype implementation on the ORBIT wireless research testbed.

Acknowledgements

Authors would like to thank King Huang and Shiwen Chen of Panasonic Solutions Development Center for their invaluable comments. This material is based upon work supported by the National Science Foundation under Grant No. ANI-0335244 and Panasonic System Solutions Development Center.

References

[1] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: A survey, Computer Networks (Elsevier) 47 (2005) 445–487.

[2] M. Ma, M.K. Denko, Y. Zhang, Wireless Quality-of-Service: Techniques, Standards and Applications, CRC Press, 2008.

[3] Y. Yang, R. Kravetz, Contention-aware admission control for ad hoc networks, IEEE Transactions on Mobile Computing 4 (4) (2005) 363–377.

[4] I.D. Chakeres, E.M. Belding-Royer, PAC: perceptive admission control for mobile wireless networks, in: Proceedings of the 1st International Conference on Quality of Service in Heterogeneous Wired/Wireless Network (QShine), Dallas, TX, 2004, pp. 18–26.

[5] G. Ahn, A.T. Campbell, A. Veres, L. Sun, SWAN: service differentiation in stateless wireless ad hoc networks, in: Proceedings of IEEE INFOCOM, vol. 2, New York, 2002, pp. 457–466.

[6] C.L.C.R. Lin, An on-demand qos routing protocol for mobile ad hoc networks, in: Proceedings of IEEE GLOBECOM, vol. 3, 2000, pp. 1783–1787.

[7] R. Prasad, C. Dovrolis, M. Murray, K. Claffy, Bandwidth estimation: metrics, measurement techniques, and tools, IEEE Network 17 (6) (2003) 27–35.

[8] M. Stemm, S. Seshan, R.H. Katz, A network measurement architecture for adaptive applications, in: Proceedings of Conference of Computer Communication, 2000, pp. 285–294.

[9] N. Hu, P. Steenkiste, Evaluation and characterization of available bandwidth probing techniques, IEEE JSAC Special Issue in Internet and WWW Measurement, Mapping, and Modeling 21 (6), (2003) 879–894.

[10] L. Chen, W.B. Heinzelman, QoS-aware routing based on bandwidth estimation for mobile ad hoc networks, IEEE Journal on Selected Areas in Communications 23 (3) (2005) 561–572.

[11] M. Deziel, L. Lamont, Implementation of an IEEE 802.11 link available bandwidth algorithm to allow cross-layering, in: Proceedings of WiMob 2005, vol. 3, Wireless and Mobile Computing, Networking and Communications, Canada, 2005, pp. 117–122.

[12] S.H. Shah, K. Chen, K. Nahrstedt, Dynamic bandwidth management in single-hop ad hoc wireless networks, Kluwer Mobile Networks and Applications 10 (2005) 199–217.

[13] K. Lakshminarayanan, V. Padmanabhan, J. Padhye, Bandwidth estimation in broadband access networks, in: Proceedings of ACM Internet Measurement Conference, Italy, 2004, pp. 314–321.

[14] T. Sun, G. Yang, L.-J. Chen, M.Y. Sanadidi, M. Gerla, A measurement study of path capacity in 802.11b based wireless networks, in: Proceedings of International Conference on Mobile Systems, Applications and Services, 2005, pp. 31–37.

[15] J. Evans, C. Filsfils, Deploying IP and MPLS QoS for multiservice networks: theory and practice, Morgan Kaufmann Publishers, San Francisco, CA, USA, 2007.

[16] L.L. Peterson, B.S. Davie, Computer Networks: A Systems Approach, third ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[17] K. Wu, J. Harms, QoS support in mobile ad hoc networks – crossing boundaries, GSA Journal of University of Alberta 1 (1) (2001) 92–106.

[18] Y. Dong, D. Makrakis, T. Sullivan, Effective admission control in multihop mobile ad hoc networks, in: Proceedings of ICCT 2003, vol. 2, 2003, pp. 1291–1294.

[19] M.G. Barry, A.T. Campbell, A. Veres, Distributed control algorithms for service differentiation in wireless packet networks, in: Proceedings of IEEE INFOCOM, Alaska, USA, 2001, pp. 582–590.

[20] D. Zhao, J. Zou, T.D. Todd, Admission control with load balancing in IEEE 802.11-based ESS mesh networks, Wireless Networks 13 (3) (2007) 351–359.

[21] H. yu Wei, K. Kim, A. Kashyap, S. Ganguly, On admission of VoIP calls over wireless mesh network, in: Proceedings of IEEE ICC, vol. 5, 2006, pp. 1990–1995.

[22] IEEE, Std. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, <http://standards.ieee.org>, 1999.

[23] B. O'Hara, A. Petrick, IEEE 802.11 Handbook: A Designer's Companion, second ed., IEEE Press, 2005.

[24] K. Fall, K. Varadha, The ns Manual, The VINT Project.

[25] X. Yang, N.H. Vaidya, On physical carrier sensing in wireless ad hoc networks, in: Proceedings of IEEE INFOCOM, vol. 4, Miami, Florida, USA, 2005, pp. 2525–2535.

[26] L. Luo, M. Gruteser, H. Liu, D. Raychaudhuri, K. Huang, S. Chen, A QoS routing and admission control scheme for 802.11 ad hoc networks, in: Proceedings of DIWANS, Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks, Los Angeles, USA, 2006, pp. 19–28.

[27] M.A. Ergin, M. Gruteser, Using packet probes for available bandwidth estimation: a wireless testbed experience, in: Proceedings of WiNTECH, International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, Los Angeles, USA, 2006, pp. 95–96.

[28] C. Dovrolis, P. Ramanathan, D. Moore, What do packet dispersion techniques measure? in: Proceedings of Conference of Computer Communication, 2001, pp. 905–914.

[29] IEEE Computer Society, Std. 802.11e: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, <http://standards.ieee.org>, 2005.

[30] MADWiFi, Multiband Atheros Driver for WiFi, <http://madwifi.org>, 2007.

[31] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, M. Singh, Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols, in: Proceedings of IEEE WCNC, vol. 3, 2005, pp. 1664–1669.

[32] M. Ott, I. Seskar, R. Siraccusa, M. Singh, ORBIT testbed software architecture: supporting experiments as a service, in: Proceedings of IEEE Tridentcom, 2005, pp. 136–145.

[33] C. Tschudin, R. Gold, O. Rensfelt, O. Wibling, LUNAR: a lightweight underlay network ad-hoc routing protocol and implementation, in: Proceedings of Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN'04), St. Petersburg, 2004.

[34] CoRe Group, Uppsala University, LUNAR: Lightweight Underlay Network Ad-hoc Routing, <http://core.it.uu.se/core/index.php/LUNAR>, 2007.

[35] TOILERS Group, Colorado School of Mines, SCORES: SCenariO characteRizEr for Simulation, <http://toilers.mines.edu/Public/SCenariO>, 2007.